

Convergent Cutting-Plane and Partial-Sampling Algorithm for Multistage Stochastic Linear Programs with Recourse¹

Z. L. CHEN² AND W. B. POWELL³

Communicated by M. Avriel

Abstract. We propose an algorithm for multistage stochastic linear programs with recourse where random quantities in different stages are independent. The algorithm approximates successively expected recourse functions by building up valid cutting planes to support these functions from below. In each iteration, for the expected recourse function in each stage, one cutting plane is generated using the dual extreme points of the next-stage problem that have been found so far. We prove that the algorithm is convergent with probability one.

Key Words. Multistage stochastic programming, cutting planes, sampling, convergence with probability one.

1. Introduction

Numerous real-world problems in applications, such as transportation (Ref. 1), production planning (Ref. 2), financial planning (Ref. 3), and many other fields (Refs. 4 and 5), can be formulated as two-stage or multistage stochastic linear programs with recourse. The characteristics of such a problem can be summarized as follows: (i) a stage usually represents a time period; (ii) the very beginning of the first stage is viewed as here and now; (iii) at the beginning of each stage, we know deterministically all the data in this stage, but know only probabilistically all the data in future stages;

¹This work was supported in part by Grant AFOSR-F49620-93-1-0098 from the Air Force Office of Scientific Research.

²Assistant Professor, Department of Systems Engineering, University of Pennsylvania, Philadelphia, Pennsylvania.

³Professor, Department of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey.

(iv) at the beginning of the first stage, decisions must be made before the realization of random data in future stages; (v) once random data in a stage becomes known, correction (i.e., recourse) actions are allowed to compensate the decisions for this stage made earlier; (vi) the goodness of the decision making is measured by the total cost, consisting of the deterministic cost in the first stage and total expected cost in the future stages.

Methods for stochastic linear programs can be generally classified into those which use a fixed sample of realizations (scenario-based methods) and those which iteratively sample realizations as the algorithm progresses (sampling-based methods).

Scenario-based methods normally approximate a stochastic problem using a relatively small set of realizations which allow the problem to be solved as a (typically large) linear program. Two-stage problems may be approximated using hundreds or in special cases thousands of scenarios, but multistage problems are normally restricted to much smaller samples. Once a set of scenarios has been generated, most scenario-based methods treat this sample as representing the entire problem which they then strive to solve to optimality. Examples of algorithms designed for this class of problems includes the diagonal quadratic approximation method of Mulvey and Ruszczyński (Ref. 6), augmented Lagrangian decomposition method of Rosa and Ruszczyński (Ref. 7), L-shaped method of Van Slyke and Wets (Ref. 8), its generalization to multistage problems by Birge (Ref. 9), and scenario aggregation method of Rockafellar and Wets (Ref. 10). All these algorithms provide optimal solutions to what are normally approximations of the original problem.

Sampling-based methods represent explicitly the complete sample space, which may be of infinite size for all practical purposes. Examples include the stochastic linearization method (Refs. 11 and 12), auxiliary function method (Ref. 13), stochastic decomposition (Ref. 14), sample path optimization (Ref. 15), and stochastic hybrid approximation method (Ref. 16). All these methods use successive samples to develop algorithms that converge in some probabilistic sense in the limit. In practical settings, statistical methods have to be used to determine convergence criteria and the solution properties after a finite number of iterations (Ref. 14).

A popular strategy to counteract the exponential growth of multistage models has been to develop successive approximations of the recourse function. It is well known (see, for example, Refs. 8 and 17) that the expected recourse function in a two-stage program can be replaced with a series of Benders cuts, where the recourse function is represented using a fixed sample. Birge (Ref. 9) extends this approach to multistage problems by proposing a nested Benders decomposition algorithm. The basic version of this method involves a forward pass through the time periods, using a specific set of cuts,

and then a backward pass, where new cuts are generated. Other authors have studied variations of this strategy (Refs. 18–22). Infanger and Morton (Ref. 23) show how the method can be extended to take advantage of interstage dependencies.

Nested Benders decomposition, as it is generally described (see for example Ref. 23), requires solving a linear program at each time period and for each scenario, where a scenario represents a full history of events up to that point in time. Let Ω_t represent the set of outcomes in time period t , and let $h_t = (\omega_1, \omega_2, \dots, \omega_t)$ represent the history of the process, where $h_t \in \mathcal{H}_t = \Omega_1 \times \Omega_2 \times \dots \times \Omega_t$ (some authors use the notation Ω_t to represent the history, whereas we use it only to denote events within a time period). Clearly, the size of \mathcal{H}_t grows exponentially with the number of stages, making nested Benders decomposition impractical for even medium-sized problems.

In this paper, we propose a new convergent algorithm for multistage stochastic linear programs with recourse that satisfy the following assumptions:

- (A1) random quantities in different stages are independent;
- (A2) the sample space of random quantities in each stage is discrete and finite;
- (A3) random quantities appear only on the right-hand side of the linear constraints in each stage;
- (A4) the feasible region of the problem in each stage is always non-empty and bounded.

As mentioned earlier, Assumption (A2) is necessary for all scenario-based methods. Assumptions (A3) and (A4) are made in many sampling-based methods including the stochastic decomposition method of Higle and Sen (Ref. 14). The nested Benders decomposition method of Birge (Ref. 9) assumes (A2) and (A3). We note that the result that we are going to present can be extended, after some refinement, to more general cases including the case where not only the right-hand side vectors, but also the matrices B_t linking neighboring stages are stochastic, and also including the case where the feasible region of the problem in each stage can be infeasible or unbounded.

Features of our method include:

- (a) At each iteration, we solve a linear program for a single realization $\omega_t \in \Omega_t$ (as opposed to each $h_t \in \mathcal{H}_t$) at each stage t . As a result, the computational requirements of the procedure per iteration grow linearly with the number of stages and the size of the sample space per stage.

- (b) We perform a simple comparison over the entire sample space Ω_t at each stage t . Thus, Ω_t may be large (say, in the tens or even hundreds of thousands), but must be finite.
- (c) Our method successively approximates the expected recourse function by building up valid cutting planes to support these functions from below.
- (d) We prove that the algorithm converges in the limit, but do not provide finite convergence.

Because we use only a partial sample [item (a) above], rather than the full sample required by other methods, we call our method cutting-plane and partial-sampling (CUPPS) algorithm. On the other hand, the full pass over the sample space in item (b) implies that the space must be finite, in contrast with true sampling techniques such as stochastic decomposition. Our method is closest to the nested Benders decomposition method of Birge (Ref. 9) and stochastic decomposition method of Higle and Sen (Ref. 14).

The research contribution of the paper is the presentation of a new algorithm for solving multistage stochastic programs, which is convergent in probability, and which is computationally tractable for problems with large numbers of outcomes per stage and large numbers of stages. The primary limitation of our method is shared by all cutting-plane algorithms, which is slow convergence when we are approximating high-dimensionality problems. Since the relative advantage of CUPPS over the classical nested Benders decomposition in terms of execution time per iteration is obvious, we do not present any numerical experiments. Our belief is that experimental work must be conducted in the context of a specific application with an algorithm that is able to take advantage of the structure of that problem.

This paper is organized as follows. In Section 2, we present the core idea of the CUPPS algorithm when applied to a two-stage problem and compare it to the L-shaped algorithm and stochastic decomposition algorithm. In Section 3, we present the details of the CUPPS algorithm. In Section 4 we give some preliminary results, and in Section 5 we establish the convergence of this algorithm. Finally, we conclude the paper in Section 6.

2. Core Idea and Comparison

In this section, we present briefly the core idea of our CUPPS method when applied to a two-stage problem and compare it to the two closest existing methods: the stochastic decomposition (SD) method of Higle and Sen (Ref. 14) and L-shaped (LS) method of Van Slyke and Wets (Ref. 8).

We begin by introducing some basic notation:

$(\Omega_t, \mathcal{F}_t, P_t)$ = probability space of the random quantities in stage t , where Ω_t is the sample space of the random quantities [hence by (A2), $|\Omega_t|$ is finite]; \mathcal{F}_t is a σ -algebra and P_t a probability measure defined over Ω_t ;
 $\Omega_t = \{\omega_{t1}, \dots, \omega_{t,q_t}\}$ = sample space of the random quantities in stage t , where $q_t = |\Omega_t|$ and ω_{ti} is a sample in Ω_t , $\forall i = 1, \dots, q_t$, with $q_1 = 1$;
 p_{ti} = probability associated with each sample $\omega_{ti} \in \Omega_t$, $\forall i = 1, \dots, q_t$, such that $\sum_{i=1}^{q_t} p_{ti} = 1$;
 \mathcal{H}_t = σ -field which represents the information available up to stage t ;
 x_t = vector of decisions in stage t ;
 c_t = vector of cost in stage t ;
 A_t = constraint matrix in stage t ;
 B_t = constraint matrix linking stage t and stage $t+1$;
 $Q_t(x_{t-1}, \omega_t) = Q_t(x_{t-1}, \omega_t | \mathcal{H}_{t-1})$ = recourse function in stage $t-1$ given the history \mathcal{H}_{t-1} ; note that Assumption (A1) guarantees the first equality;
 $\bar{Q}_t(x_{t-1}) = \sum_{i=1}^{q_t} p_{ti} Q_t(x_{t-1}, \omega_{ti}) = E_{\Omega_t} Q_t(x_{t-1}, \omega_t) = E_{\Omega_t} Q_t(x_{t-1}, \omega_t | \mathcal{H}_{t-1})$ = expected recourse function in stage $t-1$ given the history \mathcal{H}_{t-1} ; note that Assumption (A1) guarantees the second equality.

A general T -stage stochastic linear program with recourse can be formulated as follows:

$$\min \quad c_1^T x_1 + E_{\Omega_2}[\min c_2^T x_2 + \dots + E_{\Omega_T}[\min c_T^T x_T] \dots], \quad (1a)$$

$$\text{s.t.} \quad A_1 x_1 = b_1, \quad (1b)$$

$$B_1 x_1 + A_2 x_2 = \omega_2, \quad (1c)$$

$$\dots \quad \dots \quad \dots,$$

$$B_{T-1} x_{T-1} + A_T x_T = \omega_T, \quad (1d)$$

$$x_t \geq 0, \quad t = 1, \dots, T, \quad (1e)$$

$$\omega_t \in \Omega_t, \quad t = 2, \dots, T. \quad (1f)$$

For the problem of our interest that satisfies Assumption (A1), the formulation (1) can be rewritten equivalently in the following recursive form:

$$(LP_1) \quad Q_1 = \min_{x_1} \quad c_1^T x_1 + \bar{Q}_2(x_1), \quad (2)$$

$$\text{s.t.} \quad A_1 x_1 = b_1, \quad (3)$$

$$x_1 \geq 0, \quad (4)$$

where the recourse function is defined by, for $t = 2, \dots, T$,

$$(LP_t) \quad Q_t(x_{t-1}, \omega_t) = \min_{x_t} c_t^T x_t + \bar{Q}_{t+1}(x_t), \quad (5)$$

$$\text{s.t.} \quad A_t x_t = \omega_t - B_{t-1} x_{t-1}, \quad (6)$$

$$x_t \geq 0, \quad (7)$$

and $\bar{Q}_{T+1} \equiv 0$.

The core idea of the CUPPS method is to successively approximate the expected recourse function in each stage by valid cutting planes that are generated based on a known subset of dual extreme points of the next-stage problem. To be specific, let us consider the two-stage problem given by (2–7) with $T=2$. For solving this problem, each iteration k of the CUPPS algorithm involves two steps.

The first step solves an approximated problem LP_1 , which is as follows:

$$\min_{x_1} c_1^T x_1 + z, \quad (8)$$

$$\text{s.t.} \quad A_1 x_1 = b_1, \quad (9)$$

$$z + \beta_i^T x_1 \geq \alpha_i, \quad \forall i = 1, \dots, k, \quad (10)$$

$$x_1 \geq 0, \quad (11)$$

where (10) represents the k cuts generated so far. These cuts are generated in the second step and approximate the expected recourse function $\bar{Q}_2(x_1)$ by supporting it from below. Note that initially the algorithm approximates $\bar{Q}_2(x_1)$ by the first cut that is trivial, $z \geq -\infty$. Let x_1^k denote the solution to problem (8–11).

In the second step, first the algorithm randomly draws a sample, denoted as ω_2^k from Ω_2 , then solves problem LP_2 with $x_1 = x_1^k$ and $\omega_2 = \omega_2^k$. Assumption (A4) guarantees that both the optimal primal and dual solutions to this problem can always be found. Let π_k be the dual solution of this problem. Notice that, in problem LP_2 , x_1 and ω_2 appear only on the right-hand side. Thus, for any given x_1^u and x_1^v with $u \neq v$, and for given ω_{2i} , $\omega_{2j} \in \Omega_2$ with $i \neq j$, any dual extreme point of problem LP_2 with $x_1 = x_1^u$ and $\omega_2 = \omega_{2i}$ is also a dual extreme point of the problem with $x_1 = x_1^v$ and $\omega_2 = \omega_{2j}$. Let

$$\mathcal{D}^k = \{\pi_j | j = 1, \dots, k\}$$

be the set of all dual extreme points of problem LP_2 generated up to iteration k . Based on the dual extreme points in \mathcal{D}^k , the algorithm then generates a new cut,

$$z + \beta_{k+1}^T x_1 \geq \alpha_{k+1}, \quad (12)$$

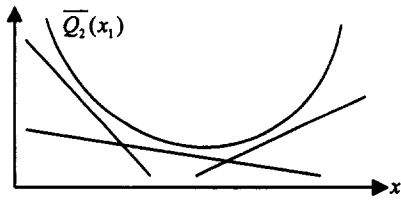
with coefficients (scalar α_{k+1} and vector β_{k+1}) given by

$$\alpha_{k+1} - \beta_{k+1}^T x_1 \equiv \sum_{j=1}^{q_2} p_{2j}(\tilde{\pi}_j)^T (\omega_{2j} - B_1 x_1), \quad (13)$$

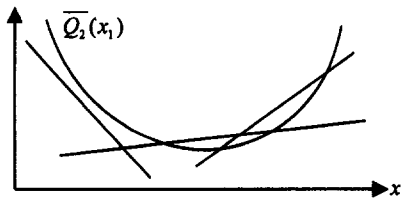
$$\tilde{\pi}_j = \operatorname{argmax}\{\pi_i^T (\omega_{2j} - B_1 x_1^k) \mid \pi_i \in \mathcal{D}^k\}. \quad (14)$$

Note that, for large problems, the summation over all possible outcomes in Ω_2 in Eq. (13) and the maximum operator applied in Eq. (14) represent the computational bottleneck of the procedure.

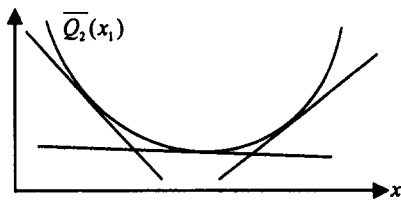
As we show in Section 4, the cut generated this way is valid for the expected recourse function $\bar{Q}_2(x_1)$, in that it supports $\bar{Q}_2(x_1)$ from below, but may not be tight, as illustrated in Fig. 1, top. The effort for generating



Cuts generated by the CUPPS method



Cuts generated by the SD method



Cuts generated by the LS method

Fig. 1. Comparison of cuts generated by the CUPPS, SD, and LS methods.

this cut involves solving one linear program (that is, problem LP_2 with $x_1 = x_1^k$, $\omega_2 = \omega_2^k$) and $O(kq_2)$ basic operations for computing α_{k+1} and β_{k+1} in (13) and (14). This cut is then added to problem (8–11). The whole procedure is then repeated.

By comparison, the SD algorithm shares similar steps, except that it not only generates new cuts, but also updates previously generated cuts. By contrast, the LS algorithm, not only solves just one problem LP_2 with $x_1 = x_1^k$ and $\omega_2 = \omega_2^k$, but instead it solves every problem LP_2 with $x_1 = x_1^k$ and $\omega_2 = \omega_{2j}$, $\forall j = 1, \dots, q_2$. Then, a new cut is generated based on the optimal dual solutions of all these q_2 problems.

Specifically, in the SD algorithm, iteration k solves problem LP_2 with $x_1 = x_1^k$ and $\omega_2 = \omega_2^k$ and generates a new cut (12) with coefficients α_{k+1} and β_{k+1} given by

$$\alpha_{k+1} - \beta_{k+1}^T x_1 \equiv \sum_{j=1}^k p_{2j} (\hat{\pi}_j)^T (\omega_2^j - B_1 x_1), \quad (15)$$

$$\hat{\pi}_j = \operatorname{argmax} \{ \pi_i^T (\omega_2^j - B_1 x_1^j) \mid \pi_i \in \mathcal{D}^j \}. \quad (16)$$

In general, the cut generated this way may not be a valid cut for the expected recourse function $\bar{Q}_2(x_1)$, as illustrated in Fig. 1 (center) because the coefficients of the cut are computed in (15) using only k samples, instead of all the samples in Ω_2 . It is easy to see that the computational effort involved here is solving one linear program and performing $O(k)$ basic operations in (15) and (16); indeed, in iteration k , one computes only $\hat{\pi}_k$, since all other $\hat{\pi}_j$, with $1 \leq j \leq k-1$, were computed earlier. Thus, the SD algorithm offers the lowest computational effort per iteration of all three algorithms.

To generate a new cut in iteration k , the LS algorithm solves every problem LP_2 with $x_1 = x_1^k$ and $\omega_2 = \omega_{2j}$, for each $j = 1, \dots, q_2$. Let u_j^k denote the optimal dual solution (obtained in iteration k) of problem LP_2 with $x_1 = x_1^k$ and $\omega_2 = \omega_{2j}$. Then, the coefficients of the new cut (12) generated by the LS algorithm in iteration k are given by

$$\alpha_{k+1} - \beta_{k+1}^T x_1 \equiv \sum_{j=1}^{q_2} p_{2j} (u_j^k)^T (\omega_{2j} - B_1 x_1). \quad (17)$$

It is easy to see that α_{k+1} and β_{k+1} given by (17) satisfy

$$\alpha_{k+1} - \beta_{k+1}^T x_1^k = \bar{Q}_2(x_1^k). \quad (18)$$

Furthermore, it is not difficult to see that the cut given by (17) is valid; hence, by (18), it is a tight cut for the expected recourse function $\bar{Q}_2(x_1)$

and touches the function at the point x_1^* , as shown in Fig. 1 (bottom). The computational effort involved here consists of solving q_2 linear programs and performing $O(q_2)$ basic operations in (17).

From the above comparison, it is quite clear that, to generate a cut, the LS algorithm needs the most computational effort, while the SD algorithm needs the least computational effort among these three algorithms. On the other hand, the quality of the cuts generated by the LS algorithm is the best in terms of their tightness. Thus, for two-stage problems, the CUPPS algorithm can be viewed as a method lying between the SD algorithm and the LS algorithm. The CUPPS algorithm attempts to build valid cuts, instead of stochastic cuts as in the SD algorithm, by using available dual extreme points that have been generated, instead of solving all the problems associated with the samples as in the LS algorithm.

For multistage problems, the core ideas of the nested Benders decomposition algorithm of Birge (Ref. 9) and of the CUPPS algorithm are similar to their respective counterparts for two-stage problems described above. Hence, we do not compare them here. The details of the CUPPS algorithm are described in Section 3. See Ref. 9 for the details of the nested Benders decomposition algorithm.

3. CUPPS Algorithm

In each iteration, the CUPPS algorithm solves an approximated problem LP_t , denoted as AP_t for each $t = 1, 2, \dots, T-1$, and a problem LP_T . In problem AP_t , the expected recourse function $\bar{Q}_{t+1}(x_t)$ is approximated by some cuts that support it from below. After solving problem AP_t or LP_T , the algorithm generates a cut that is valid for the expected recourse function $\bar{Q}_t(x_{t-1})$ and adds this cut to problem AP_{t-1} . In the course of the algorithm, the approximated problems AP_t , for $t = 1, \dots, T-1$, approximate the original LP_t more and more accurately.

In each iteration, the algorithm generates one cut for each expected recourse function $\bar{Q}_t(x_{t-1})$ for $t = 2, \dots, T$. At the very beginning, the algorithm uses the following initial cut to support the function $\bar{Q}_t(x_{t-1})$ from below:

$$z_{t+1} \geq -\infty. \quad (19)$$

Certainly, this is a valid cut. Thus, there are a total of $k+1$ cuts in problem AP_t right after iteration k .

Suppose that, right after iteration $k-1$, the approximated problems AP_1 and AP_t , for $t=2, \dots, T-1$, are as follows:

$$(AP_1) \quad \hat{Q}_1^k = \min_{x_1, z_2} c_1^T x_1 + z_2, \quad (20)$$

$$\text{s.t.} \quad A_1 x_1 = b_1, \quad (21)$$

$$z_2 + (\beta_2^i)^T x_1 \geq \alpha_{2i}, \quad \forall i = 1, \dots, k, \quad (22)$$

$$x_1 \geq 0; \quad (23)$$

$$(AP_t) \quad \hat{Q}_t^k(x_{t-1}, \omega_t) = \min_{x_t, z_{t+1}} c_t^T x_t + z_{t+1}, \quad (24)$$

$$\text{s.t.} \quad A_t x_t = \omega_t - B_{t-1} x_{t-1}, \quad (25)$$

$$z_{t+1} + (\beta_{t+1}^i)^T x_t \geq \alpha_{t+1,i}, \quad \forall i = 1, \dots, k, \quad (26)$$

$$x_t \geq 0, \quad (27)$$

where (26) represents k cuts that have been generated up to iteration $k-1$ for the expected recourse function $\bar{Q}_{t+1}(x_t)$. We describe later how these cuts are generated.

Then, in the next iteration (i.e., iteration k), the CUPPS algorithm first solves problem AP_1 . Let the primal and dual solutions be denoted by (x_1^k, z_2^k) and (π_1^k, ρ_1^k) , where π_1^k and ρ_1^k are the vectors representing the dual solutions corresponding to (21) and (22). Next, for each $t=2, \dots, T$ in this order, first the algorithm draws a sample (denoted by ω_t^k) from Ω_t , then solves problem AP_t for $t < T$ (or problem LP_T for $t = T$), with $x_{t-1} = x_{t-1}^k$ and $\omega_t = \omega_t^k$, and gets the primal and dual solutions, denoted by (x_t^k, z_{t+1}^k) and (π_t^k, ρ_t^k) (or by x_T^k and π_T^k), where π_t^k and ρ_t^k are the vectors representing the dual solutions corresponding to (25) [or (6) in problem LP_T] and (26).

For $t=2, \dots, T$, denote $\mathcal{D}_t^k = \text{set of all the dual extreme points generated so far right after iteration } k \text{ for problem } AP_t, \text{ for } t < T, \text{ or problem } LP_T \text{ for } t = T$. Then, after problem AP_t (or LP_T) is solved in iteration $k-1$, this set of dual extreme points is updated by, for $t=2, \dots, T-1$,

$$\mathcal{D}_t^k = \mathcal{D}_t^{k-1} \cup \{(\pi_t^k, \rho_t^k)\}, \quad (28)$$

and for $t=T$,

$$\mathcal{D}_T^k = \mathcal{D}_T^{k-1} \cup \{\pi_T^k\}. \quad (29)$$

Note that elements in the set \mathcal{D}_t^k may have different dimensions. Elements generated earlier have smaller dimensions than those generated later.

Throughout this paper, whenever we calculate the inner product of or compare two vectors with different dimensions, we assume that the vector with smaller dimension is extended by attaching zeros to it such that it has the same dimension as the other vector. We show in Section 5 that any element in \mathcal{D}_i^k generated earlier than iteration k , if extended accordingly by attaching zeros to it, is still a dual extreme point of problem AP_i formed later than iteration k .

Based on the dual extreme points in the set \mathcal{D}_i^k that have been generated so far, the algorithm then generates a new cut [i.e., the $(k+1)$ th cut] for the function $\bar{Q}_i(x_{i-1})$. This cut is given by

$$z_i + (\beta_i^{k+1})^T x_{i-1} \geq \alpha_{i,k+1}, \quad (30)$$

with

$$\beta_i^{k+1} = \sum_{i=1}^{q_i} p_{ii} B_{i-1}^T \pi_i(x_{i-1}^k, \omega_{ii}, \mathcal{D}_i^k), \quad 2 \leq i \leq T, \quad (31)$$

$$\alpha_{i,k+1} = \sum_{i=1}^{q_i} p_{ii} [\omega_{ii}^T \pi_i(x_{i-1}^k, \omega_{ii}, \mathcal{D}_i^k) + (\alpha_{i+1}^k)^T \rho_i(x_{i-1}^k, \omega_{ii}, \mathcal{D}_i^k)], \quad (32)$$

$$2 \leq i \leq T-1,$$

$$\alpha_{T,k+1} = \sum_{i=1}^{q_T} p_{Ti} (\omega_{Ti})^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k), \quad (33)$$

where, for any $\omega_i \in \Omega_i$ and $2 \leq i \leq T-1$,

$$\begin{aligned} & (\pi_i(x_{i-1}^k, \omega_i, \mathcal{D}_i^k), \rho_i(x_{i-1}^k, \omega_i, \mathcal{D}_i^k)) \\ & = \operatorname{argmax} \{ \pi_i^T (\omega_i - B_{i-1} x_{i-1}^k) + \rho_i^T \alpha_{i+1}^k \mid (\pi_i, \rho_i) \in \mathcal{D}_i^k \}, \end{aligned} \quad (34)$$

and for any $\omega_T \in \Omega_T$,

$$\pi_T(x_{T-1}^k, \omega_T, \mathcal{D}_T^k) = \operatorname{argmax} \{ \pi_T^T (\omega_T - B_{T-1} x_{T-1}^k) \mid \pi_T \in \mathcal{D}_T^k \}, \quad (35)$$

and for $2 \leq i \leq T$, α_i^k is the vector defined as

$$\alpha_i^k = (\alpha_{i,1}, \dots, \alpha_{i,k}). \quad (36)$$

This new cut (30) is added to the preceding problem AP_{i-1} . Then, the algorithm moves forward to solve the next problem AP_{i+1} . Similarly, a new cut is generated and added to the preceding problem AP_i . Finally, the algorithm solves problem LP_T , generates a new cut, and adds this cut to problem AP_{T-1} . This ends iteration k .

Now we are ready to give the details of the CUPPS algorithm.

CUPPS Algorithm.

- Step 0. For $t = 1, 2, \dots, T-1$, formulate the initial approximated problem AP_t with only one initial cut given by (19). Let the set of dual extreme points $\mathcal{D}_t^1 = \emptyset$ for all $t = 2, 3, \dots, T$. Set the iteration counter $k = 1$.
- Step 1. Solve problem AP_1 . Get the optimal primal solution (x_1^k, z_2^k) and optimal solution value \hat{Q}_1^k .
- Step 2. For $t = 2, \dots, T-1$ in this order, do the following. Draw a random sample ω_t^k from Ω_t . For given $x_{t-1} = x_{t-1}^k$ and $\omega_t = \omega_t^k$, solve problem AP_t . Get the optimal primal solution (x_t^k, z_{t+1}^k) , optimal solution value $\hat{Q}_t^k(x_{t-1}^k, \omega_t^k)$, and optimal dual solution (π_t^k, ρ_t^k) . Get the latest set of dual extreme points \mathcal{D}_t^k by (28). Generate the $(k+1)$ th cut given by (30). Add this cut to problem AP_{t-1} .
- Step 3. Draw a random sample ω_T^k from Ω_T . For $x_{T-1} = x_{T-1}^k$ and $\omega_T = \omega_T^k$, solve problem LP_T . Get the optimal primal solution x_T^k , optimal solution value $Q_T(x_{T-1}^k, \omega_T^k)$, and optimal dual solution π_T^k . Get the latest set of dual extreme points \mathcal{D}_T^k by (29). Generate the $(k+1)$ th cut given by (30) with $t = T$. Add this cut to problem AP_{T-1} .
- Step 4. Set $k = k + 1$; go to Step 1.

4. Preliminary Results

In this section, we prove that all the cuts generated in the CUPPS algorithm for the expected recourse function $\bar{Q}_t(x_{t-1})$, $\forall t = 2, \dots, T$, are valid; that is, they support $\bar{Q}_t(x_{t-1})$ from below. Also, we give two basic results that are used in Section 5.

Define, for any $k \geq 1$ and $1 \leq t \leq T-1$,

$$\hat{\bar{Q}}_t^k(x_{t-1}) = \sum_{i=1}^{q_t} p_{ti} \hat{Q}_t^k(x_{t-1}, \omega_{ti}).$$

Lemma 4.1. In the CUPPS algorithm, each cut added to problem AP_{T-1} is a valid cut for the function $\bar{Q}_T(x_{T-1})$.

Proof. Clearly, in problem AP_{T-1} , the very first cut given by (19) with $t = T-1$ is valid for $\bar{Q}_T(x_{T-1})$. Now, consider the k th cut in AP_{T-1} for any $k \geq 2$. Clearly, for any given x_{T-1} and ω_{Ti} , for any $i = 1, \dots, q_T$, $\pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k)$ is a dual feasible solution to problem LP_T with a given

x_{T-1} and $\omega_T = \omega_{Tl}$. This implies that, for any x_{T-1} and $i = 1, \dots, q_T$,

$$Q_T(x_{T-1}, \omega_{Tl}) \geq (\omega_{Tl} - \beta_{T-1} x_{T-1})^T \pi_T(x_{T-1}^k, \omega_{Tl}, \mathcal{D}_T^k).$$

First multiplying by p_{Tl} , then summing over all i on the both sides, we have

$$\begin{aligned} \bar{Q}_T(x_{T-1}) &\geq \sum_{i=1}^{q_T} p_{Ti} (\omega_{Ti} - \beta_{T-1} x_{T-1})^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k) \\ &= \sum_{i=1}^{q_T} p_{Ti} (\omega_{Ti})^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k) \\ &\quad - \sum_{i=1}^{q_T} p_{Ti} x_{T-1}^T \beta_{T-1}^T \pi_T(x_{T-1}^k, \omega_{Ti}, \mathcal{D}_T^k) \\ &= \alpha_{T,k+1} - (\beta_T^{k+1})^T x_{T-1}. \end{aligned}$$

This shows that the k th cut in AP_{T-1} is valid for $\bar{Q}_T(x_{T-1})$. \square

Lemma 4.2. In the CUPPS algorithm, for any t with $1 \leq t \leq T-1$, and for any k with $k \geq 1$, every element in the set \mathcal{D}_t^k is a dual extreme point of problem AP_t formed right after any iteration j with $j \geq k-1$.

Proof. First, it is easy to see that any element (π_t, ρ_t) in the set \mathcal{D}_t^k is a dual extreme point of problem AP_t generated in some iteration i with $i \leq k$. Let DP1 denote the dual of problem AP_t formed right after iteration $i-1$. Then, (π_t, ρ_t) is an extreme point of problem DP1. Let DP2 denote the dual of problem AP_t formed right after iteration j for any given $j \geq k$. Then, in order to prove the lemma, we only need to show that (π_t, ρ_t) , if extended by adding a proper number of zeros to its end, is also an extreme point of problem DP2.

Since the algorithm adds one cut to problem AP_t in each iteration, and once a cut is added it will always be there, problems DP1 and DP2 are the same, except that in DP2 there are $j-i$ more columns. Thus, (π_t, ρ_t) is feasible to DP2 if we extend it by adding $j-i$ zeros to its end. Denote this extended vector by (π_t, ρ_t^+) , where

$$(\rho_t^+)_l = (\rho_t)_l, \quad l = 1, \dots, i, \quad (37a)$$

$$(\rho_t^+)_l = 0, \quad l = i+1, \dots, j. \quad (37b)$$

In the following, we show that (π_t, ρ_t^+) is an extreme point of DP2. We prove it by contradiction. If it is not, then there exist two different solutions of DP2, denoted by (π_t^1, ρ_t^1) and (π_t^2, ρ_t^2) , such that

$$(\pi_t, \rho_t^+) = (1/2)(\pi_t^1, \rho_t^1) + (1/2)(\pi_t^2, \rho_t^2). \quad (38)$$

Clearly,

$$(\rho_l^1)_l \geq 0, \quad (\rho_l^2)_l \geq 0, \quad l = 1, \dots, j.$$

Hence, by (37), we have that

$$(\rho_l^1)_l = (\rho_l^2)_l = 0, \quad l = i+1, \dots, j.$$

Now, define two vectors $\hat{\rho}_i^1$ and $\hat{\rho}_i^2$ with dimension i such that

$$(\hat{\rho}_i^1)_l = (\rho_l^1)_l, \quad (\hat{\rho}_i^2)_l = (\rho_l^2)_l, \quad l = 1, \dots, i.$$

Then, the fact that $(\pi_i^1, \rho_i^1) \neq (\pi_i^2, \rho_i^2)$ implies that

$$(\pi_i^1, \hat{\rho}_i^1) \neq (\pi_i^2, \hat{\rho}_i^2), \quad (39)$$

and (38) implies that

$$(\pi_i, \rho_i) = (1/2)(\pi_i^1, \hat{\rho}_i^1) + (1/2)(\pi_i^2, \hat{\rho}_i^2). \quad (40)$$

On the other hand, it is easy to show that both $(\pi_i^1, \hat{\rho}_i^1)$ and $(\pi_i^2, \hat{\rho}_i^2)$ are feasible for DP1. Then, (39) and (40) are in contradiction with the fact that (π_i, ρ_i) is an extreme point of DP1. This shows that (π_i, ρ_i^+) must be an extreme point of DP2. \square

Lemma 4.3. In the CUPPS algorithm, for any $\tau = 1, \dots, T-1$, each cut added to problem AP_τ is a valid cut for the function $\bar{Q}_{\tau+1}(x_\tau)$.

Proof. We prove this by induction on τ . For $\tau = T-1$, this result has been proved in Lemma 4.1. Now, assume that, for any given $t < T-1$, this result is true for $\tau = t$. We need to prove that this result is also true for $\tau = t+1$. For any given $k \geq 1$, it is easy to see that, right after iteration k , problem AP_t is equivalent to the following problem:

$$\min_{x_t} \{c_t^T x_t + F_{t+1}^k(x_t) \mid (25), (27)\}, \quad (41)$$

where $F_{t+1}^k(x_t)$ is a piecewise linear function formed by the k cuts in (26). By the induction assumption, these cuts are valid for the function $\bar{Q}_{t+1}(x_t)$; then,

$$F_{t+1}^k(x_t) \leq \bar{Q}_{t+1}(x_t), \quad \text{for any } x_t.$$

On the other hand, the feasible region of problem (41) is the same as that of problem LP_t . So, it must be true that the optimal objective function value of problem (41) is no more than that of problem LP_t ; i.e., for any x_{t-1} and ω_t ,

$$\hat{Q}_t^k(x_{t-1}, \omega_t) \leq Q_t(x_{t-1}, \omega_t).$$

Taking expectation of both sides, we have

$$\hat{\bar{Q}}_t^k(x_{t-1}) \leq \bar{Q}_t(x_{t-1}). \quad (42)$$

On the other hand, by Lemma 4.2, any element in \mathcal{D}_t^k is a dual extreme point of problem AP_t in iteration k . Thus, for any $\omega_{it} \in \Omega_t$, with $1 \leq i \leq q_t$, the vector $(\pi_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k), \rho_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k))$ is a dual extreme point of problem AP_t in iteration k . This means that, for any x_{t-1} and $i = 1, \dots, q_t$,

$$\begin{aligned} \hat{\bar{Q}}_t^k(x_{t-1}, \omega_{it}) &\geq (\omega_{it} - B_{t-1}x_{t-1})^T \pi_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k) \\ &\quad + (\alpha_{t+1}^k)^T \rho_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k). \end{aligned}$$

First multiplying by p_{it} , then summing over all i on the both sides, we have

$$\begin{aligned} \hat{\bar{Q}}_t^k(x_{t-1}) &\geq \sum_{i=1}^{q_t} p_{it} [(\omega_{it} - B_{t-1}x_{t-1})^T \pi_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k) \\ &\quad + (\alpha_{t+1}^k)^T \rho_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k)] \\ &= - \sum_{i=1}^{q_t} p_{it} (\pi_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k))^T B_{t-1} x_{t-1} \\ &\quad + \sum_{i=1}^{q_t} p_{it} [\omega_{it}^T \pi_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k) + (\alpha_{t+1}^k)^T \rho_t(x_{t-1}^k, \omega_{it}, \mathcal{D}_t^k)] \\ &= -(\beta_t^{k+1})^T x_{t-1} + \alpha_{t,k+1}. \end{aligned}$$

Together with (42), this gives

$$\bar{Q}_t(x_{t-1}) \geq \alpha_{t,k+1} - (\beta_t^{k+1})^T x_{t-1}.$$

This shows that the $(k+1)$ th cut in problem AP_{t-1} , generated right after problem AP_t is solved in iteration k , is a valid cut for the function $\bar{Q}_t(x_{t-1})$. This shows that the result is true when $\tau = t-1$. Therefore, by induction, we have shown the lemma. \square

Lemma 4.4. The following properties all hold:

- (a) $Q_1 \geq \hat{Q}_1^{k+1} \geq \hat{Q}_1^k$, for any $k \geq 1$;
- (b) $\bar{Q}_t(x_{t-1}, \omega_t) \geq \hat{\bar{Q}}_t^{k+1}(x_{t-1}, \omega_t) \geq \hat{\bar{Q}}_t^k(x_{t-1}, \omega_t)$, for any $k \geq 1$, $\omega_t \in \Omega_t$, x_{t-1} , and $2 \leq t \leq T$;
- (c) $\bar{Q}_t(x_{t-1}) \geq \hat{\bar{Q}}_t^{k+1}(x_{t-1}) \geq \hat{\bar{Q}}_t^k(x_{t+1})$, for any $k \geq 1$, x_{t-1} , and $2 \leq t \leq T$.

Proof. These results are straightforward from Lemmas 4.1 and 4.3. Thus, we omit the proofs for them. \square

Remark 4.1. If we run infinitely many iterations of the CUPPS algorithm, then with probability one, each particular sample ω_{it} , for any $i = 1, \dots, q_t$ and $t = 2, \dots, T$, will be drawn infinitely many times.

Proof. In one iteration of the CUPPS algorithm, exactly one sample is drawn from Ω_t , for each $t = 2, \dots, T$. For any given t and i , with $2 \leq t \leq T$ and $1 \leq i \leq q_t$, the probability that the sample ω_{it} is drawn out in one iteration is $p_{it} > 0$. If we run infinitely many iterations of the algorithm, then by the well-known Borel–Cantelli lemma (see e.g. Ref. 24), the probability that ω_{it} is drawn out for infinitely many times is one. This shows the result. \square

Remark 4.2. Given any LP of the form $\min\{c^T x \mid Ax = b, x \geq 0\}$, define \mathcal{B} to be the set of all possible right-hand side vectors b such that there exists an optimal solution to the LP with $b \in \mathcal{B}$. Define a function $f(b)$ over the set \mathcal{B} , to be the optimal objective value of the LP with $b \in \mathcal{B}$. Then, the function f is continuous in the set \mathcal{B} ; i.e., for any given $\epsilon > 0$ and $b \in \mathcal{B}$, there is a $\delta > 0$ such that

$$|f(b) - f(h)| < \epsilon, \quad \text{for any } h \in \mathcal{B} \text{ with } \|b - h\| < \delta.$$

5. Limiting Behavior of the Algorithm

In this section, we analyze the limiting behavior of the CUPPS algorithm and prove that the algorithm is convergent with probability one (wpl). First, Lemma 5.1 provides bounds on certain convergent sequences. Then, we prove convergence using a classical inductive proof. Lemma 5.2 demonstrates the convergence of the solution value of problem AP_{T-1} to that of problem LP_{T-1} . Finally, Lemma 5.3 shows that the result is true, by induction, for all remaining stages. The heart of our proof is contained in Lemma 5.2, while the inductive proof in Lemma 5.3 is similar in style to that of Lemma 5.2.

Lemma 5.1. For any given infinite subset \mathcal{J} of $\mathcal{N} = \{1, 2, \dots\}$, if the sequence of vectors $\{x_{T-1}^k\}_{k \in \mathcal{J}}$ converges to some vector x_{T-1}^0 , then for any given $\epsilon > 0$:

- (a) there exists an integer v_1 such that, for any dual extreme point π_T of problem LP_T ,

$$|\pi_T^T B_{T-1}(x_{T-1}^l - x_{T-1}^m)| < \epsilon/6, \text{ for any } l, m > v_1 \text{ and } l, m \in \mathcal{J}; \quad (43)$$

- (b) there exists an integer v_{2i} , for every $1 \leq i \leq q_T$, such that, for any $l, m > v_{2i}$ and $l, m \in \mathcal{J}$,

$$|Q_T(x_{T-1}^m, \omega_{Ti}) - Q_T(x_{T-1}^0, \omega_{Ti})| < \epsilon/12; \quad (44)$$

furthermore, let $v_2 = \max\{v_{21}, v_{22}, \dots, v_{2q_T}\}$; for any $l, m > v_2$ and $l, m \in \mathcal{J}$, and $1 \leq i \leq q_T$, we have

$$|Q_T(x_{T-1}^l, \omega_{Ti}) - Q_T(x_{T-1}^m, \omega_{Ti})| < \epsilon/6; \quad (45)$$

(c) there exists an integer v_3 such that, for any $m > v_3$ and $m \in \mathcal{J}$,

$$|Q_{T-1}(x_{T-2}^m, \omega_{T-1}^0) - Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)| < \epsilon/2. \quad (46)$$

Proof.

(a) Assume that we are given a dual extreme point π_T of problem LP_T . By Assumption (A4), we can assume $\|\pi_T\| < Y$ for some finite positive number Y . The convergence of the sequence $\{x_{T-1}^k\}_{k \in \mathcal{J}}$ implies that the sequence $\{B_{T-1}x_{T-1}^k\}_{k \in \mathcal{J}}$ is also convergent. Thus, for any given $\epsilon > 0$, there exists some integer v_1 such that, for any $l, m > v_1$ and $l, m \in \mathcal{J}$, we have

$$\|B_{T-1}x_{T-1}^l - B_{T-1}x_{T-1}^m\| < \epsilon/6Y,$$

which implies that, $\forall l, m > v_1$ and $l, m \in \mathcal{J}$,

$$|\pi_T^T B_{T-1}(x_{T-1}^l - x_{T-1}^m)| \leq \|\pi_T\| \cdot \|B_{T-1}x_{T-1}^l - B_{T-1}x_{T-1}^m\| < \epsilon/6.$$

This shows (a).

(b) By Remark 4.2, for any given ω_T , the function $Q_T(x_{T-1}, \omega_T)$ is continuous in x_{T-1} . For any i with $1 \leq i \leq q_T$, consider the function $Q_T(x_{T-1}, \omega_{Ti})$ at point x_{T-1}^0 . By the continuity of this function, for any $\epsilon > 0$, there exists $\delta_i > 0$ such that, if $\|x_{T-1} - x_{T-1}^0\| < \delta_i$, then

$$|Q_T(x_{T-1}, \omega_{Ti}) - Q_T(x_{T-1}^0, \omega_{Ti})| < \epsilon/12.$$

On the other hand, since $\{x_{T-1}^k\}_{k \in \mathcal{J}}$ is convergent, then for a given δ_i , there exists an integer v_{2i} such that, for any $m > v_{2i}$ and $m \in \mathcal{J}$, we have

$$\|x_{T-1}^m - x_{T-1}^0\| < \delta_i.$$

All this implies (44), which further implies that, $\forall l, m > v_{2i}$ and $l, m \in \mathcal{J}$,

$$\begin{aligned} & |Q_T(x_{T-1}^l, \omega_{Ti}) - Q_T(x_{T-1}^m, \omega_{Ti})| \\ & \leq |Q_T(x_{T-1}^l, \omega_{Ti}) - Q_T(x_{T-1}^0, \omega_{Ti})| \\ & \quad + |Q_T(x_{T-1}^0, \omega_{Ti}) - Q_T(x_{T-1}^m, \omega_{Ti})| \\ & < \epsilon/12 + \epsilon/12 = \epsilon/6. \end{aligned}$$

Let $v_2 = \max\{v_{21}, v_{22}, \dots, v_{2q_T}\}$. Then, we have the result (45). This shows (b).

(c) Similarly, the function $Q_{T-1}(x_{T-2}, \omega_{T-1}^0)$ is continuous in x_{T-2} . We can use a similar argument to prove (46). \square

Lemma 5.2. For any given infinite subset \mathcal{K} of $\mathcal{N} = \{1, 2, \dots\}$, if the conditions below are satisfied:

- (i) $\omega_{T-1}^k = \omega_{T-1}^0$, for some given $\omega_{T-1}^0 \in \Omega_{T-1}$ for any $k \in \mathcal{K}$;
- (ii) the sequence of vectors $\{x_{T-2}^k\}_{k \in \mathcal{K}}$ converges to some given vector x_{T-2}^0 .

Then, there exists an infinite subset \mathcal{J} of \mathcal{K} such that:

- (a) the sequence $\{x_{T-1}^k\}_{k \in \mathcal{J}}$ converges to some vector x_{T-1}^0 ;
- (b) the sequence $\{\bar{Q}_{T-1}^k(x_{T-2}^0, \omega_{T-1}^0)\}_{k \in \mathcal{J}}$ converges wpl to $Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)$;
- (c) the sequence $\{z_T^k\}_{k \in \mathcal{J}}$ converges wpl to $\bar{Q}_T(x_{T-1}^0)$.

Proof. For any given infinite subset \mathcal{K} of \mathcal{N} , suppose that the conditions (i) and (ii) of the lemma are satisfied.

(a) In iteration $k \in \mathcal{K}$, for given $x_{T-2} = x_{T-2}^k$ and $\omega_{T-1}^k = \omega_{T-1}^0$, the algorithm solves problem AP_{T-1} and gets the solution (x_{T-1}^k, z_T^k) . By Assumption (A4), the sequence $\{x_{T-1}^k\}_{k \in \mathcal{K}}$ is bounded. Thus, there must exist an infinite subset \mathcal{J} of \mathcal{K} such that the sequence of vectors $\{x_{T-1}^k\}_{k \in \mathcal{J}}$ converges to some vector x_{T-1}^0 . This proves (a).

(b) By Lemma 5.1, for given $\epsilon > 0$, there exist integers v_1, v_{2i} for $1 \leq i \leq q_T$, and v_3 such that (43)–(46) hold. Define

$$v_2 = \max\{v_{21}, v_{22}, \dots, v_{2q_T}\}, \quad v = \max\{v_1, v_2\}.$$

Partition the set \mathcal{J} into q_T subsets $\mathcal{J}_1, \mathcal{J}_2, \dots, \mathcal{J}_{q_T}$ such that, for $i = 1, 2, \dots, q_T$, $\omega_T^k = \omega_{Ti}$, $\forall k \in \mathcal{J}_i$. In other words, \mathcal{J}_i is the set of iteration indices where the sample ω_{Ti} is drawn out for ω_T . Since \mathcal{J} is infinite, by Remark 4.1, in the iterations $k \in \mathcal{J}$, any particular sample $\omega_{Ti} \in \Omega_T$ can be drawn infinitely many times wpl, which means that every set \mathcal{J}_i , for $1 \leq i \leq q_T$, is infinite wpl.

For each $1 \leq i \leq q_T$, let r_i be the first element in \mathcal{J}_i that is greater than v . Now, consider the algorithm in iterations r_1, r_2, \dots, r_{q_T} . In iteration r_i , the algorithm solves problem LP_T with $x_{T-1} = x_{T-1}^{r_i}$ and $\omega_T = \omega_{Ti}$. This generates the optimal dual solution $\pi_T^{r_i}$ of LP_T . So, for $1 \leq i \leq q_T$,

$$Q_T(x_{T-1}^{r_i}, \omega_{Ti}) = (\pi_T^{r_i})^T (\omega_{Ti} - B_{T-1} x_{T-1}^{r_i}). \quad (47)$$

Let r be the first element in \mathcal{J} that is greater than $\max\{r_1, r_2, \dots, r_{q_T}\}$. Clearly, before iteration r , the algorithm has already generated the dual extreme point $\pi_T^{r_i}$ satisfying (47), $\forall i = 1, 2, \dots, q_T$. After problem LP_T , with $x_{T-1} = x_{T-1}^r$ and some ω_T , is solved in iteration r , we get the set \mathcal{D}_T^r that contains the dual extreme points of problem LP_T generated so far. Then,

the $(r+1)$ th cut for the function $\bar{Q}_T(x_{T-1})$ is generated and added to problem AP_{T-1} . This cut is given by

$$z_T + (\beta_T^{r+1})^T x_{T-1} \geq \alpha_{T,r+1}, \quad (48)$$

where the vector β_T^{r+1} and the scalar $\alpha_{T,r+1}$ are defined in (31) with $t=T$ and $k=r$ and in (33) with $k=r$; that is,

$$\beta_T^{r+1} = \sum_{i=1}^{q_T} p_{Ti} B_{T-1}^T \pi_T(x_{T-1}^i, \omega_{Ti}, \mathcal{D}_T^i), \quad (49)$$

$$\alpha_{T,r+1} = \sum_{i=1}^{q_T} p_{Ti} (\omega_{Ti})^T \pi_T(x_{T-1}^i, \omega_{Ti}, \mathcal{D}_T^i), \quad (50)$$

where

$$\begin{aligned} & \pi_T(x_{T-1}^i, \omega_{Ti}, \mathcal{D}_T^i) \\ &= \operatorname{argmax} \{ \pi_T^T (\omega_{Ti} - B_{T-1} x_{T-1}^i) \mid \pi_T \in \mathcal{D}_T^i \}. \end{aligned} \quad (51)$$

It is easy to see that

$$\pi_T^i \in \mathcal{D}_T^i, \quad \forall i=1, 2, \dots, q_T.$$

Thus, (51) implies that, for each $i=1, 2, \dots, q_T$,

$$\begin{aligned} & (\pi_T(x_{T-1}^i, \omega_{Ti}, \mathcal{D}_T^i))^T (\omega_{Ti} - B_{T-1} x_{T-1}^i) \\ & \geq (\pi_T^i)^T (\omega_{Ti} - B_{T-1} x_{T-1}^i). \end{aligned} \quad (52)$$

Now, let

$$s = \max\{v_3, r\}.$$

Let N be the first element in \mathcal{J} that is greater than s . Consider any iteration n , with $n > N$ and $n \in \mathcal{J}$. In iteration n , the algorithm solves problem AP_{T-1} , with $x_{T-2} = x_{T-2}^n$ and $\omega_{T-1} = \omega_{T-1}^0$, and gets the solution value $\hat{Q}_{T-1}^n(x_{T-2}^n, \omega_{T-1}^0)$ and the solution (x_{T-1}^n, z_T^n) . Note that, since $n > r$, in iteration n the cut (48) is already in problem AP_{T-1} . Hence, the solution (x_{T-1}^n, z_T^n) must satisfy (48), that is,

$$\begin{aligned} z_T^n & \geq \alpha_{T,r+1} - (\beta_T^{r+1})^T x_{T-1}^n \\ &= \sum_{i=1}^{q_T} p_{Ti} [(\pi_T(x_{T-1}^i, \omega_{Ti}, \mathcal{D}_T^i))^T (\omega_{Ti} - B_{T-1} x_{T-1}^i)] \\ &= \sum_{i=1}^{q_T} p_{Ti} [(\pi_T(x_{T-1}^i, \omega_{Ti}, \mathcal{D}_T^i))^T (\omega_{Ti} - B_{T-1} x_{T-1}^i)] \\ & \quad + \sum_{i=1}^{q_T} p_{Ti} [(\pi_T(x_{T-1}^i, \omega_{Ti}, \mathcal{D}_T^i))^T B_{T-1} (x_{T-1}^i - x_{T-1}^n)]. \end{aligned} \quad (53)$$

For notational convenience, we define

$$\Delta_1 = \sum_{i=1}^{q_T} p_{T_i} [(\pi_T(x'_{T-1}, \omega_{T_i}, \mathcal{D}_T))^T B_{T-1} (x'_{T-1} - x''_{T-1})],$$

$$\Delta_2 = \sum_{i=1}^{q_T} p_{T_i} [(\pi'_T)^T B_{T-1} (x'_{T-1} - x'_{T-1})],$$

$$\Delta_3 = \sum_{i=1}^{q_T} p_{T_i} [Q_T(x'_{T-1}, \omega_{T_i}) - Q_T(x''_{T-1}, \omega_{T_i})].$$

By (47) and (52), Inequality (53) implies

$$\begin{aligned} z_T^n &\geq \sum_{i=1}^{q_T} p_{T_i} [(\pi'_T)^T (\omega_{T_i} - B_{T-1} x'_{T-1})] + \Delta_1 \\ &= \sum_{i=1}^{q_T} p_{T_i} [(\pi'_T)^T (\omega_{T_i} - B_{T-1} x'_{T-1})] + \Delta_1 + \Delta_2 \\ &= \sum_{i=1}^{q_T} p_{T_i} Q_T(x'_{T-1}, \omega_{T_i}) + \Delta_1 + \Delta_2 \\ &= \sum_{i=1}^{q_T} p_{T_i} Q_T(x''_{T-1}, \omega_{T_i}) + \Delta_1 + \Delta_2 + \Delta_3 \\ &= \bar{Q}_T(x''_{T-1}) + \Delta_1 + \Delta_2 + \Delta_3. \end{aligned} \quad (54)$$

Inequality (54) implies

$$\begin{aligned} \hat{Q}_{T-1}^n(x''_{T-2}, \omega_{T-1}^0) &= c_{T-1}^T x''_{T-1} + z_T^n \\ &\geq c_{T-1}^T x''_{T-1} + \bar{Q}_T(x''_{T-1}) + \Delta_1 + \Delta_2 + \Delta_3. \end{aligned} \quad (55)$$

It is easy to see that $x_{T-1} = x''_{T-1}$ is a feasible solution to problem LP_{T-1} , with $x_{T-2} = x''_{T-2}$ and $\omega_{T-1} = \omega_{T-1}^0$, which means that

$$Q_{T-1}(x''_{T-2}, \omega_{T-1}^0) \leq c_{T-1}^T x''_{T-1} + \bar{Q}_T(x''_{T-1}). \quad (56)$$

Combining (55) and (56), we get

$$\hat{Q}_{T-1}^n(x''_{T-2}, \omega_{T-1}^0) \geq Q_{T-1}(x''_{T-2}, \omega_{T-1}^0) + \Delta_1 + \Delta_2 + \Delta_3. \quad (57)$$

On the other hand, by Lemma 4.4,

$$\hat{Q}_{T-1}^n(x''_{T-2}, \omega_{T-1}^0) \leq Q_{T-1}(x''_{T-2}, \omega_{T-1}^0). \quad (58)$$

From (57) and (58), it is not difficult to show that

$$\begin{aligned} |\hat{Q}_{T-1}^n(x''_{T-2}, \omega_{T-1}^0) - Q_{T-1}(x''_{T-2}, \omega_{T-1}^0)| \\ \leq |\Delta_1| + |\Delta_2| + |\Delta_3|. \end{aligned} \quad (59)$$

This gives

$$\begin{aligned}
 & |\hat{Q}_{T-1}^n(x_{T-2}^n, \omega_{T-1}^0) - Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)| \\
 & \leq |\hat{Q}_{T-1}^n(x_{T-2}^n, \omega_{T-1}^0) - Q_{T-1}(x_{T-2}^n, \omega_{T-1}^0)| \\
 & \quad + |Q_{T-1}(x_{T-2}^n, \omega_{T-1}^0) - Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)| \\
 & \leq |\Delta_1| + |\Delta_2| + |\Delta_3| \\
 & \quad + |Q_{T-1}(x_{T-2}^n, \omega_{T-1}^0) - Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)|.
 \end{aligned} \tag{60}$$

It is easy to see that

$$n > r > r_i > \max\{v_1, v_2\}, \quad i = 1, 2, \dots, q_T.$$

Hence, by (43) and (45), we have

$$|\Delta_1| < \epsilon/6, \quad |\Delta_2| < \epsilon/6, \quad |\Delta_3| < \epsilon/6. \tag{61}$$

Similarly, since $n > v_3$, by (46) we have

$$|Q_{T-1}(x_{T-2}^n, \omega_{T-1}^0) - Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)| < \epsilon/2.$$

Thus, (60) and (61) imply

$$\begin{aligned}
 & |\hat{Q}_{T-1}^n(x_{T-2}^n, \omega_{T-1}^0) - Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)| \\
 & < \epsilon/2 + \epsilon/6 + \epsilon/6 + \epsilon/6 = \epsilon.
 \end{aligned}$$

This means that the sequence $\{\hat{Q}_{T-1}^k(x_{T-2}^k, \omega_{T-1}^0)\}_{k \in \mathcal{J}}$ converges to $Q_{T-1}(x_{T-2}^0, \omega_{T-1}^0)$. Since in the proof we have used the result that each \mathcal{J}_i is infinite wpl, this convergence is wpl. This shows part (b) of the lemma.

(c) The convergence of the sequence $\{z_T^k\}_{k \in \mathcal{J}}$ can be proved similarly as follows. The relations (55) and (58) imply

$$c_{T-1}^T x_{T-1}^n + z_T^n \leq Q_{T-1}(x_{T-2}^n, \omega_{T-1}^0).$$

Then, by (56), we have

$$z_T^n \leq \bar{Q}_T(x_{T-1}^n).$$

This, together with (54), implies

$$|z_T^n - \bar{Q}_T(x_{T-1}^n)| \leq |\Delta_1| + |\Delta_2| + |\Delta_3|. \tag{62}$$

By (44), we have

$$\begin{aligned}
 & |\bar{Q}_T(x_{T-1}^n) - \bar{Q}_T(x_{T-1}^0)| \\
 &= \left| \sum_{i=1}^{q_T} p_{Ti} (Q_T(x_{T-1}^n, \omega_{Ti}) - Q_T(x_{T-1}^0, \omega_{Ti})) \right| \\
 &\leq \sum_{i=1}^{q_T} p_{Ti} |Q_T(x_{T-1}^n, \omega_{Ti}) - Q_T(x_{T-1}^0, \omega_{Ti})| \\
 &\leq \epsilon/12 < \epsilon/2.
 \end{aligned}$$

This, together with (61) and (62), implies

$$\begin{aligned}
 |z_T^n - \bar{Q}_T(x_{T-1}^0)| &\leq |z_T^n - \bar{Q}_T(x_{T-1}^n)| + |\bar{Q}_T(x_{T-1}^n) - \bar{Q}_T(x_{T-1}^0)| \\
 &< |\Delta_1| + |\Delta_2| + |\Delta_3| + \epsilon/2 \\
 &< \epsilon.
 \end{aligned}$$

Hence, the sequence $\{z_T^k\}_{k \in \mathcal{J}}$ converges wpl to $\bar{Q}_T(x_{T-1}^0)$. This shows part (c) of the lemma. \square

Now, we demonstrate convergence of all other stages.

Lemma 5.3. For any given τ , $1 \leq \tau \leq T-2$, and any given infinite subset \mathcal{K} of $\mathcal{N} = \{1, 2, \dots\}$, if the conditions below are satisfied:

- (i) $\omega_\tau^k = \omega_\tau^0$ for some given $\omega_\tau^0 \in \Omega_\tau$, for any $k \in \mathcal{K}$;
- (ii) the sequence of vectors $\{x_{\tau-1}^k\}_{k \in \mathcal{K}}$ converges to some given vector $x_{\tau-1}^0$.

Then, there exists an infinite subset \mathcal{J} of \mathcal{K} such that:

- (a) the sequence $\{x_\tau^k\}_{k \in \mathcal{J}}$ converges to some vector x_τ^0 ;
- (b) the sequence $\{\hat{Q}_\tau^k(x_{\tau-1}^k, \omega_\tau^0)\}_{k \in \mathcal{J}}$ converges wpl to $Q_\tau(x_{\tau-1}^0, \omega_\tau^0)$;
- (c) the sequence $\{z_{\tau+1}^k\}_{k \in \mathcal{J}}$ converges wpl to $\bar{Q}_{\tau+1}(x_\tau^0)$.

Proof. We prove the lemma by induction on τ . When $\tau = T-1$, this lemma is exactly Lemma 5.1 and hence holds. Suppose that this lemma holds when $\tau = t$. We need to prove that it also holds when $\tau = t-1$. The proof technique is similar to that of Lemma 5.2. Thus, we provide only a sketch of proof.

For any given infinite subset \mathcal{K} of \mathcal{N} , suppose that, when $\tau = t-1$, conditions (i) and (ii) of the lemma are satisfied.

(a) In iteration $k \in \mathcal{K}$, the algorithm solves problem AP_{t-1} , with $x_{t-2} = x_{t-2}^k$ and $\omega_{t-1} = \omega_{t-1}^0$, and gets the solution (x_{t-1}^k, z_t^k) . By Assumption (A4), the sequence $\{x_{t-1}^k\}_{k \in \mathcal{K}}$ is bounded. Thus, there must exist an infinite subset \mathcal{L} of \mathcal{K} such that the sequence of vectors $\{x_{t-1}^k\}_{k \in \mathcal{L}}$ converges to some vector x_{t-1}^0 .

Partition the set \mathcal{L} into q_t subsets $\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_{q_t}$ such that, for $i = 1, 2, \dots, q_t$,

$$\omega_t^k = \omega_{it}, \quad \forall k \in \mathcal{L}_i.$$

In other words, \mathcal{L}_i is the set of iteration indices where the sample ω_{it} is drawn out for ω_t . By Remark 4.1, it is easy to see that every set \mathcal{L}_i , for $1 \leq i \leq q_t$, is infinite wpl.

For each $i = 1, 2, \dots, q_t$, by the induction assumption that the lemma holds for $\tau = t$ and by the facts that $\omega_t^k = \omega_{it}$, for all $k \in \mathcal{L}_i$, and that the sequence $\{x_{t-1}^k\}_{k \in \mathcal{L}}$ [hence, the sequence $\{x_{t-1}^k\}_{k \in \mathcal{L}_i}$] converges to some vector x_{t-1}^0 , there must exist a subset \mathcal{J}_i of \mathcal{L}_i , for each $i = 1, 2, \dots, q_t$, such that the sequence $\{\hat{Q}_t^k(x_{t-1}^k, \omega_{it})\}_{k \in \mathcal{J}_i}$ converges wpl to $Q_t(x_{t-1}^0, \omega_{it})$. Thus, it is easy to show that, for any given $\epsilon > 0$, there exists an integer u_i such that, wpl,

$$|\hat{Q}_t^m(x_{t-1}^m, \omega_{it}) - Q_t(x_{t-1}^0, \omega_{it})| < \epsilon/12, \quad \forall m > u_i \text{ and } m \in \mathcal{J}_i. \quad (63)$$

Define

$$\mathcal{J} = \bigcup_{i=1}^{q_t} \mathcal{J}_i.$$

Clearly, $\mathcal{J} \subseteq \mathcal{L}$. Hence, the sequence $\{x_{t-1}^k\}_{k \in \mathcal{J}}$ converges to x_{t-1}^0 . This shows that part (a) of the lemma holds when $\tau = t-1$.

(b) First, using (63) and logic similar to that in the proof of Lemma 5.1, we can get the following results for any given $\epsilon > 0$:

- (i) there exists an integer v_1 such that, for any dual extreme point (π_t, ρ_t) of problem AP_t ,

$$|\pi_t^T B_{t-1}(x_{t-1}^l - x_{t-1}^m)| < \epsilon/6, \quad \forall l, m > v_1 \text{ and } l, m \in \mathcal{J}; \quad (64)$$

- (ii) there exists an integer v_{2i} , for every $1 \leq i \leq q_t$, such that

$$|Q_t(x_{t-1}^m, \omega_{it}) - Q_t(x_{t-1}^0, \omega_{it})| < \epsilon/12, \\ \forall l, m > v_{2i} \text{ and } l, m \in \mathcal{J}; \quad (65)$$

furthermore, let $v_2 = \max\{u_1, u_2, \dots, u_{q_t}, v_{21}, v_{22}, \dots, v_{2q_t}\}$; we have

$$|\hat{Q}_t^l(x_{t-1}^l, \omega_{it}) - Q_t(x_{t-1}^m, \omega_{it})| < \epsilon/6, \\ \forall l, m > v_2 \text{ and } l, m \in \mathcal{J}, \text{ and } \forall 1 \leq i \leq q_t; \quad (66)$$

(iii) there exists an integer v_3 such that

$$\begin{aligned} |Q_{t-1}(x_{t-2}^m, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| &< \epsilon/2, \\ \forall m > v_3 \text{ and } m \in \mathcal{J}. \end{aligned} \quad (67)$$

Let us define v, r_i for $1 \leq i \leq q_t, r, s$, and N exactly in the same way as in the proof of Lemma 5.2. Consider the algorithm in iterations r_i and r . Similar to (47) and (52), the following relations hold for each $i = 1, 2, \dots, q_t$:

$$\hat{Q}_{t-1}^{r_i}(x_{t-1}^{r_i}, \omega_{tt}) = (\pi_{t-1}^{r_i})^T(\omega_{tt} - B_{t-1}x_{t-1}^{r_i}) + (\rho_{t-1}^{r_i})^T\alpha_{t+1}^{r_i}, \quad (68)$$

$$\begin{aligned} &(\pi_t(x_{t-1}^r, \omega_{tt}, \mathcal{D}_t^r))^T(\omega_{tt} - B_{t-1}x_{t-1}^r) + (\rho_t(x_{t-1}^r, \omega_{tt}, \mathcal{D}_t^r))^T\alpha_{t+1}^r \\ &\geq (\pi_{t-1}^{r_i})^T(\omega_{tt} - B_{t-1}x_{t-1}^{r_i}) + (\rho_{t-1}^{r_i})^T\alpha_{t+1}^{r_i}. \end{aligned} \quad (69)$$

Now, consider any iteration n and $n > N$ and $n \in \mathcal{J}$. In iteration n , the algorithm solves problem AP_{t-1} , with given $x_{t-2} = x_{t-2}^n$ and $\omega_{t-1} = \omega_{t-1}^0$, and gets the solution value $\hat{Q}_{t-1}^n(x_{t-2}^n, \omega_{t-1}^0)$ and the solution (x_{t-1}^n, z_t^n) . Since $n > r$, in iteration n the $(r+1)$ th cut, with coefficients β_{t-1}^{r+1} and $\alpha_{t,r+1}$ defined by (31) and (32), is already in problem AP_{t-1} . Hence, the solution (x_{t-1}^n, z_t^n) must satisfy that cut; that is,

$$\begin{aligned} z_t^n &\geq \alpha_{t,r+1} - (\beta_{t-1}^{r+1})^T x_{t-1}^n \\ &= \sum_{i=1}^{q_t} p_{ti} [(\pi_t(x_{t-1}^r, \omega_{tt}, \mathcal{D}_t^r))^T(\omega_{tt} - B_{t-1}x_{t-1}^n) \\ &\quad + (\alpha_{t+1}^r)^T \rho_t(x_{t-1}^r, \omega_{tt}, \mathcal{D}_t^r)]. \end{aligned}$$

By (68) and (69), we can get a result similar to (54) as follows:

$$z_t^n \geq \bar{Q}_t(x_{t-1}^n) + \Delta_1 + \Delta_2 + \Delta_3, \quad (70)$$

where

$$\begin{aligned} \Delta_1 &= \sum_{i=1}^{q_t} p_{ti} (\pi_t(x_{t-1}^r, \omega_{tt}, \mathcal{D}_t^r))^T B_{t-1} (x_{t-1}^r - x_{t-1}^n), \\ \Delta_2 &= \sum_{i=1}^{q_t} p_{ti} (\pi_{t-1}^{r_i})^T B_{t-1} (x_{t-1}^{r_i} - x_{t-1}^r), \\ \Delta_3 &= \sum_{i=1}^{q_t} p_{ti} [\hat{Q}_{t-1}^{r_i}(x_{t-1}^{r_i}, \omega_{tt}) - Q_t(x_{t-1}^n, \omega_{tt})]. \end{aligned}$$

Using the same argument as in the proof of Lemma 5.2, we can then show the following result that is similar to (60):

$$\begin{aligned} & |\hat{Q}_{t-1}^n(x_{t-2}^n, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| \\ & \leq |Q_{t-1}(x_{t-2}^n, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| + |\Delta_1| + |\Delta_2| + |\Delta_3|. \end{aligned} \quad (71)$$

By (64) and (66), we have

$$|\Delta_1| < \epsilon/6, \quad |\Delta_2| < \epsilon/6, \quad |\Delta_3| < \epsilon/6. \quad (72)$$

Similarly, by (67), we have

$$|Q_{t-1}(x_{t-2}^n, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| < \epsilon/2.$$

Thus, (71) implies

$$|\hat{Q}_{t-1}^n(x_{t-2}^n, \omega_{t-1}^0) - Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)| < \epsilon.$$

This means that the sequence $\{\hat{Q}_{t-1}^k(x_{t-2}^k, \omega_{t-1}^0)\}_{k \in \mathcal{J}}$ converges to $Q_{t-1}(x_{t-2}^0, \omega_{t-1}^0)$. Since in the proof we have used some results that are true wpl, this convergence is wpl. This shows that part (b) of the lemma holds when $\tau = t - 1$.

(c) The convergence of the sequence $\{z_t^k\}_{k \in \mathcal{J}}$ can be proved similarly to part (c) of Lemma 5.2. We do not give any details here.

Therefore, by induction, we have proved the lemma. \square

Theorem 5.1. The sequence of solution values $\{\hat{Q}_1^k\}_{k \in \mathcal{N}}$ of problem AP_1 converges wpl to the optimal value Q_1 .

Proof. In the approximated problem AP_1 , we can view the constraint $A_1x_1 = b_1$ as $A_1x_1 = \omega_1 - B_0x_0$, with $\omega_1 \equiv b_1$, $x_0 \equiv 0$, and any given B_0 ; and we can view the value \hat{Q}_1^k as the function $\hat{Q}_1^k(x_0, \omega_1)$. Thus, when $\tau = 1$ and $\mathcal{X} = \mathcal{N}$, the conditions (i) and (ii) of Lemma 5.3 are certainly satisfied. Applying Lemma 5.3, we have that there exists an infinite subset \mathcal{J} of \mathcal{N} , such that the sequence $\{\hat{Q}_1^k\}_{k \in \mathcal{J}}$ converges wpl to $Q_1(b_1, 0) \equiv Q_1$.

On the other hand, by Lemma 4.4, the sequence $\{\hat{Q}_1^k\}_{k \in \mathcal{N}}$ is nondecreasing. We know that, if a monotone sequence has a convergent subsequence that converges to some value, then the whole sequence must converge to that value. Therefore, the sequence $\{\hat{Q}_1^k\}_{k \in \mathcal{N}}$ converges wpl to Q_1 . \square

Theorem 5.2. Any accumulation point of the sequence $\{x_1^k\}_{k \in \mathcal{N}}$ is an optimal solution wpl of problem LP_1 .

Proof. To prove this, we need to show only that any convergent subsequence of the sequence $\{x_1^k\}_{k \in \mathcal{N}}$ converges wpl to an optimal solution of problem LP_1 .

Consider a subsequence \mathcal{K} of \mathcal{N} such that the sequence $\{x_1^k\}_{k \in \mathcal{K}}$ converges to some vector x_1^0 . With the identification $b_0 = \omega_1 - B_0 x_0$ as in the proof of Theorem 5.1, applying Lemma 5.3, we can show that there exists a subsequence \mathcal{J} of \mathcal{K} , such that the sequence $\{z_2^k\}_{k \in \mathcal{J}}$ converges wpl to $\bar{Q}_2(x_1^0)$.

On the other hand, we know that

$$\hat{Q}_1^k = c_1^T x_1^k + z_2^k, \quad \forall k \in \mathcal{N}.$$

Thus,

$$z_2^k = \hat{Q}_1^k - c_1^T x_1^k, \quad \forall k \in \mathcal{N}. \quad (73)$$

Now, in the set \mathcal{K} , we take the limit on both sides of (73). Since the sequence $\{x_1^k\}_{k \in \mathcal{K}}$ converges to x_1^0 , and since by Theorem 5.1 the sequence $\{\hat{Q}_1^k\}_{k \in \mathcal{K}}$ converges wpl to Q_1 , then the sequence $\{z_2^k\}_{k \in \mathcal{K}}$, converges wpl to $Q_1 - c_1^T x_1^0$. We know that, if a convergent sequence has a subsequence that converges to some value, then the whole sequence converges to that value, so the following must be true:

$$\bar{Q}_2(x_1^0) = Q_1 - c_1^T x_1^0.$$

Hence,

$$Q_1 = c_1^T x_1^0 + \bar{Q}_2(x_1^0). \quad (74)$$

Since x_1^k is a feasible solution to problem LP_1 , for any $k \in \mathcal{K}$, then by Assumption (A4), the limit x_1^0 of the sequence $\{x_1^k\}_{k \in \mathcal{K}}$ must be a feasible solution to problem LP_1 . Therefore, (74) implies that the solution x_1^0 is actually optimal to problem LP_1 . This shows the theorem. \square

6. Conclusions

In this paper, we have proposed the CUPPS algorithm, a sampling-based algorithm, for solving multistage stochastic linear programs. We have proved that the algorithm is convergent with probability one.

We believe that multistage stochastic linear programs are much harder than two-stage ones. It is unlikely that a scenario-based algorithm is capable of solving a multistage problem with the sample space in each stage containing 1000 samples. For such a problem with T as small as 3, there are 10^6 scenarios. Standard methods, such as the diagonal quadratic approximation method of Mulvey and Ruszczyński (Ref. 6) and augmented Lagrangian decomposition method of Rosa and Ruszczyński (Ref. 7), that reformulate the stochastic problems as a deterministic equivalence, are certainly incapable of dealing with such a problem. We also doubt that the nested Benders

decomposition algorithm of Birge (Ref. 9) can handle such a problem, because one iteration alone involves solving at least 3000 linear programs. We believe that sampling-based methods that require only to solve a small number of linear programs in each iteration are more likely to be successful in solving multistage stochastic linear programs involving a large number of samples in each stage. The CUPPS algorithm proposed in this paper is the first such method that is convergent for multistage stochastic linear programs.

References

1. POWELL, W. B., JAILLET, P., and ODONI, A., *Stochastic and Dynamic Networks and Routing*, Handbook in Operations Research and Management Science, Volume on Networks, Edited by C. Monma, T. Magnanti, and M. Ball, North Holland, Amsterdam, Holland, pp. 141–295, 1995.
2. ESCUDERO, L. F., KAMESAM, P. V., KING, A. J., and WETS, R., *Production Planning via Scenario Modeling*, Annals of Operations Research, Vol. 43, pp. 311–335, 1993.
3. MULVEY, J. M., and VLADIMIROU, H., *Stochastic Network Programming for Financial Planning Problems*, Management Science, Vol. 38, pp. 1642–1664, 1992.
4. BIRGE, J. R., *Stochastic Programming Computation and Applications*, INFORMS Journal on Computing, Vol. 9, pp. 111–133, 1997.
5. BIRGE, J. R., and MULVEY, J. M., *Stochastic Programming in Industrial Engineering*, Technical Report, Department of Industrial and Operations Engineering, University of Michigan, 1996.
6. MULVEY, J. M., and RUSZCZYNSKI, A., *A Diagonal Quadratic Approximation Method for Large-Scale Linear Programs*, Operations Research Letters, Vol. 12, pp. 205–215, 1991.
7. ROSA, C., and RUSZCZYNSKI, A., *On Augmented Lagrangian Decomposition Methods for Multistage Stochastic Programs*, Annals of Operations Research, Vol. 64, pp. 289–309, 1996.
8. VAN SLYKE, R., and WETS, R., *L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming*, SIAM Journal on Applied Mathematics, Vol. 17, pp. 638–663, 1969.
9. BIRGE, J. R., *Decomposition and Partitioning Techniques for Multistage Stochastic Linear Programs*, Operations Research, Vol. 33, pp. 989–1007, 1985.
10. ROCKAFELLAR, T. R., and WETS, R., *Scenarios and Policy Aggregation in Optimization under Uncertainty*, Mathematics of Operations Research, Vol. 16, pp. 119–147, 1991.
11. GUPAL, A. M., and BAZHENOV, L. G., *A Stochastic Method of Linearization*, Cybernetics, Vol. 8, pp. 482–484, 1972.
12. ERMOLIEV, Y., *Stochastic Quasigradient Methods and Their Application to System Optimization*, Stochastics, Vol. 9, pp. 1–36, 1983.

13. CULIOLI, J. C., and COHEN, G., *Decomposition/Coordination Algorithms in Stochastic Optimization*, SIAM Journal on Control and Optimization, Vol. 28, pp. 1372–1403, 1990.
14. HIGLE, J. L., and SEN, S., *Stochastic Decomposition: An Algorithm for Two-Stage Linear Programs with Recourse*, Mathematics of Operations Research, Vol. 16, pp. 650–669, 1991.
15. ROBINSON, S. M., *Analysis of Sample Path Optimization*, Mathematics of Operations Research, Vol. 21, pp. 1–528, 1996.
16. CHEUNG, R. K. M., and POWELL, W. B., *A Stochastic Hybrid Approximation Procedure, with an Application to Dynamic Networks*, Technical Report SOR-94-02, Department of Civil Engineering and Operations Research, Princeton University, 1994.
17. WETS, R., *Programming under Uncertainty: The Solution Set*, SIAM Journal on Applied Mathematics, Vol. 14, pp. 1143–1151, 1966.
18. WITTRICK, R. J., *Advances in a Nested Decomposition Algorithm for Solving Staircase Linear Programs*, Technical Report SOL-83-2, Stanford University, 1983.
19. GASSMAN, H. I., *MSLIP: A Computer Code for the Multistage Stochastic Linear Programming Problem*, Mathematical Programming, Vol. 47, pp. 407–423, 1990.
20. PEREIRA, M. V. F., and PINTO, L. M. V. G., *Multistage Stochastic Optimization Applied to Energy Planning*, Mathematical Programming, Vol. 52, pp. 359–375, 1991.
21. INFANGER, G., *Planning under Uncertainty: Solving Large-Scale Stochastic Linear Programs*, Boyd and Fraser, Scientific Press Series, New York, New York, 1994.
22. MORTON, D. P., *An Enhanced Decomposition Algorithm for Multistage Stochastic Hydroelectric Scheduling*, Annals of Operations Research, Vol. 64, pp. 211–235, 1996.
23. INFANGER, G., and MORTON, D. P., *Cut Sharing for Multistage Stochastic Linear Programs with Interstate Dependency*, Mathematical Programming, Vol. 75, pp. 241–256, 1996.
24. CHUNG, K. L., *A Course in Probability Theory*, Academic Press, New York, New York, 1974.